High Performance scientific computing
Assignment 1

Anshuman kumar
Roll no. 120010036

February 3, 2016

1. The serial portion of an algorithm constitutes 20 %.

   (a) What is the maximum achievable speedup for this algorithm? [5pts]

   (b) If the desirable speedup is 50. What should be maximum percentage of the serial portion for the algorithm? [5pts]

Solution:
(a) Using Amdahl's law

$$S(N) = \frac{1}{(1-P) + \frac{P}{N}}$$

We got the maxinmum speed up when n = ∞
We get Speed up = 1/0.2 = 5.0

(b) The speed up needed is 50.0. Using the Amdahl's law we get

$$\frac{1}{(1-P)} = 50.0$$

We get 1-P = 0.02
So finally we we should have maximum of 2% Serial Code

2. . Consider a hypothetical Von Neumann type machine shown in the figure. This machine has a cache of 1 KiloByte and a main memory of 10 MegaBytes. The cost of fetching one floating point number (8 bytes in size) from the cache is 1 CPU cycle (or one clock tick). In the event the data needed is not in the cache, a chunk of data 1 KiloByte in size is fetched from the main memory to cache, replacing the contents of the cache. The cost of fetching data from main memory is 150 CPU cycles.

Now, considering the following snippet code, where matrix A is initialised to some integer values

```
#define N 1023
sum = 0;
for( i=0; i<=N; i++ )
for( j=0; j<=N; j++ )
sum = sum + A[i][j];
```

(a) What is the cost in CPU cycles to fetch the requisite data for the matrix A to perform the computation (code is in C)? Disregard the cost for the variable sum [20pts]

(b) What will be the cost again in terms of CPU cycles, if the exact same code is written in FORTRAN? [20pts]

Solution:

(a) C is row-major order language.
Size of int = 4bytes
Max No of int in cache = 1024/4 = 256

For Internal loop:
No of times cpu accesing main memory = 1024/256 = 4
No of times cpu accessing cache = 512 (Taking into account it can transfer 8 bytes at one cpu cycle)
Total no of CPU cycle = 512 + 150*4 = 1124

No of external for loop = 1024
Total No of cpu cycles = 1024* 1124 = 1150976

(b) Fortan is column major order language.So for taking a sum it has to access the memory again and again.Everytime it has to access main memory for getting a number.

No of times cpu accesses main memory = 1024* 1024 = 1048576
No of times cpu accesses cache = 1024* 1024 = 1048576
Total Cpu cycle = 1048576*150 + 1048576 = 158334976